OriginZero Job Interview

İlk 4 soruyu Türkçe/İngilizce yanıtlayabilirsiniz. 5. ve 6. soruların cevapları İngilizce beklenmektedir. Ekip arkadaşlarımızla tanışmayı dört gözle bekliyoruz. Görüşmek üzere.

1. **What is the IoT project that you want to create if you had the chance. If you had a chance to create an IoT project, what would it be?**

*İmkanınız olsa hayata geçirmek istediğiniz bir IoT projesi var mı? Varsa nedir?*

I would like to create an IoT system for forests. I want to create a system that collects environmental data from forest to preserve nature. In past, I focused on forest fires. I designed and implemented a system that could help prediction and early response. I would have like to extend it even more to cover plants and possibly some animals.

1. **What do you think are the 3 most important qualities that a code must have? Ex. Sustainability, readability, reliability, testability, having less code line etc.**

*Bir kodun sahip olması gereken en önemli 3 özellik sizce nedir? Örnek; Sürdürülebilirlik, okunabilirlik, güvenilirlik, teste yatkınlık, daha az satıra sahip olması vb.*

* Readability. I like progression, so I don’t like wasting time for things preventing me from progression. Messy code makes it harder to understand and waste time. Unreadable code is usually taking more time than rewriting it properly and methodically. If I or someone else organized that code, the time could put into progression. That wastes everyone’s time. If the code is rewritten, the time original writer put into that code is wasted, because code is different now. Overall, it’s inefficient use of time.
* Robustness. I am not sure if robustness is the exact word for this. I almost exclusively code for embedded devices. Embedded devices get exposed to unexpected situations and these situations may result in errors or hang ups. Since embedded devices interact with physical world, these malfunctions can easily result in physical harm. This might not only cause financial harm but also hurt living organism.
* Scalability. Code should be easily improved. It should always allow for some degree of expansion, this could be an update (feature, bugfix, security etc.) or simply extending scope of the project (like increasing number of sensors or actuators). Lack of scalability leads to recreating a code that does the similar or same thing in numbers, and creates convoluted project. For private projects this is an important aspect, because it makes it easy to continue project after first stages.

1. **What is "Low code”? What are the differences between “Low Code” and “No Code”? Do you think it contributes to automation process? If so, what are they?**

*Low code nedir? No-code ile arasındaki fark nedir? Otomasyon sürecine bir katkısı olduğunu düşünüyor musunuz? Varsa sizce ne gibi katkıları var?*

Low code is a concept to make programming more accessible to nonprofessionals and/or decrease development time for professionals. Traditional coding requires significant amount of knowledge and effort to achieve things, and generally complicated text-based development environments. On the other hand, low code is usually done with a graphical gui and all the tedious steps of building an application is previously done. Low code generally only requires user to code how the data flows and what operations done on it. This is main distinction between low code and no code. No code aims to completely eliminate any kind of coding or operations. Low code or no code hugely help automation process.

I can’t quote where I heard it from but I’ve heard that NASA used to use LabView Software to control space telescopes. Strictly speaking LabView is not completely low code program, but it’s fitting for example. Kinematics, and any other required calculations modelled by space professionals, in LabView by drag and drop blocks. The LabView software served mathematicians and aerospace engineers as a Low code platform. As more down to earth example the farmers can set up automated feeding system with no code platform. Even a farmer with no knowledge of programming can automate their job, which helps making automation more common. Finally, related to IoT, Azure offers low code or no code platform, with central management.

1. **How can you ensure the security of the serial communication protocols? Why?**

*Seri haberleşme protokollerinde güvenliği nasıl sağlarsınız? Neden?*

We can ensure security of serial communication protocols. Secure communication is not a new problem in technology and it has been solved before. There is no need to reinvent the wheel. The fast way would be using symmetrical encryption, where encryption and decryption is done with a single identical key. This process works best if both parties have secure way of storing key. The more secure way of encryption is asymmetric encryption. In asymmetric encryption to keys are used to encrypt or decrypt the message public and private keys. Private keys can be thought of decryption key, and public is encryption key. The sender uses receivers public key to encrypt his message and receiver uses their private key to decrypt it. So only receiver can understand message. This process is usually slower than symmetric encryption, and in embedded systems memory can be read protected so as long as receiving end also store key securely symmetric is more desirable.

1. **Think about 3 hazards & risks that can be encountered by customer while using real time industrial IoT device and explain software solutions for these issues.**

For error data collection for an industrial process (eg. QA application), for some reason, edge device can stop working. In that situation if an error occurs or emergency cannot trigger and give alarm. Which can lead to disaster. This can easily be prevented by periodical pings or self-checks to central server, that the device is up and running. If these checks not received in expected interval central server can give warning to user about nonoperational edge node.

IoT implementations can face issues with limited or no connectivity. In urban areas connectivity is easy to receive, but in rural areas connectivity is usually not present. This is a big inconvenience, because strength of IoT automation is controlling from afar. Ideally, we don’t want to move near nodes to control. This problem can be solved by mesh networks. For example, an IoT automation for irrigation. Since fields are usually away from urban areas internet connectivity can be poor or partially covered. Instead of making each node requiring powerful modem for connectivity, nodes can create mesh network and connect one server that is connected to internet with powerful modem. For more robust and serverless solution, some of the edge nodes equipped with powerful modem.

Depending on IoT implementation security is big concern. Men in the middle attacks can be weaponized for some situations. Data from IoT devices can be used against companies or malicious towards users. It can even harm physically, either by actuators or by faking data (eg. health monitor data). To prevent this type of attacks all the communications should be done encrypted. The way of doing it is explained in question before. Additionally, alongside software some tamper protection would reinforce hardware security.

1. **What is System Identification Methodology in Control Systems? Compare at least 2 estimation models such as Nonlinear and State-Space. Do you ever used Matlab System Identification Toolbox?**

I have never used System Identification Toolbox, and had limited knowledge about this topic, but I’ve gathered this information from my research. System Identification Methodology is creating model of system’s essential dynamics with captured output data from known input. It varies slightly from just fitting a curve onto output data by including some relation from previous data so that it captures more of dynamics of the system. Kind of similar to Moore state machines, it uses both input, and previous state/output to calculate output.

Compared to transfer function techniques, state-space system is advantageous for multi-input and output systems and usability among all types of systems (linear or non-linear). It is also easier to run with computers because it can benefit from raw discrete computation power of a PC.

1. **Please look at the attach file which named “question.c”**

* Problem was at line 37 while toggling LED. The code increased “button\_no” before toggling LED. So, when button 2 press detected LED 3 toggled instead of LED 2.
* In line 7, function prototype missing a parameter. Shouldn’t create any problems but not a good practice.
* Rather than waiting for next acknowledge time, last acknowledge time is stored and checked if difference from counter bigger than acknowledge interval defined at the top of the code. Acknowledgement is not guaranteed to be every 100 ms, this might be problem. Because no specifics given for this, I tried to keep the code as similar as the original code.
* Counter overflowing is something to consider. Changing it to unsinged integer should keep system working without issues, but if things start to go wrong after 50 days this could be reason.